

Branch: master ▾

node-qt / README.md

Find file

Copy path

 arturadib readme

63d6279 on 14 Apr 2012

1 contributor

166 lines (81 sloc) 4.26 KB

Node-Qt

Node-Qt provides native bindings to the [Qt library](#) as a [Node.js addon](#). The focus is on graphics and audio bindings; there is no need to duplicate the functionality of the Node API and its modules.

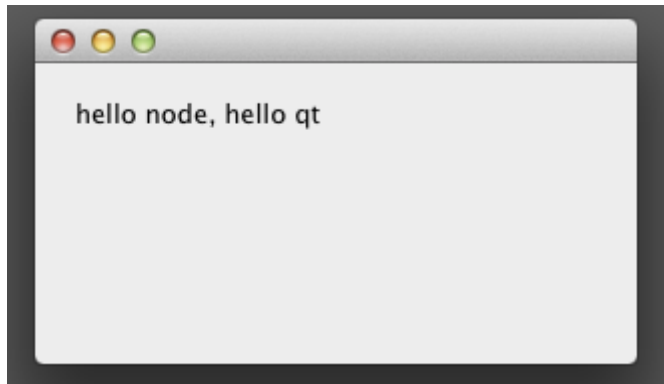
We try to follow [Qt's API](#) as closely as possible, but sometimes quirks are inevitable (for example, virtual methods that handle events are translated into callback setters). See the header files in `src/` for a list of available bindings, and comments in `.cc` files for possible API differences.

For a translation of Qt's APIs into HTML5 APIs, see [Node-Five](#).

Supported platforms: **Mac OS X** | **Windows** | **Linux**

Hello world

Ever wanted to create native apps directly from Node? Here's a simple example illustrating how to create a native window via `QWidget()` and draw via `QPainter()` :



```
var qt = require('node-qt'),
    app = new qt.QApplication,
    window = new qt.QWidget;

// Prevent objects from being GC'd
global.app = app;
global.window = window;

// Quirk: the virtual method paintEvent() is mapped into a callback setter
window.paintEvent(function() {
  var p = new qt.QPainter();
  p.begin(window);
  p.drawText(20, 30, 'hello node, hello qt');
  p.end();
});

window.resize(300, 150);
window.show();

// Join Node's event loop
setInterval(app.processEvents, 0);
```

Getting started

From your project directory, run (see below for requirements):

```
$ npm install node-qt
```

Alternatively, to use the latest development version from Github:

```
$ npm install git://github.com/arturadib/node-qt.git
```

This will download and build Node-Qt in `node_modules/`. Then create a new file, say `helloworld.js`, copy the example above and run Node as usual:

```
$ node helloworld
```

See the [examples/](#) directory for other simple use cases.

Build requirements

Node-Qt was designed to build seamlessly with minimal dependencies on most platforms. The necessary platform-dependent Qt binaries are bundled with the module (due to heterogeneous dependencies, Linux is an exception).

For all platforms: Node \geq **0.6.14**

- **Mac:** Python, Make, and GCC.
- **Windows:** Python and MSVC++ (either [free](#) or commercial).
- **Linux:** Python, Make, GCC, pkg-config, and Qt 4.7+. To install pkg-config and Qt on Ubuntu: `$ sudo apt-get install pkg-config qt-sdk`.

Contributing

Building and testing

To download and build the development version:

```
$ git clone git://github.com/arturadib/node-qt.git
$ cd node-qt
$ npm install
```

To run the unit tests:

```
$ node make test
```

(Ignore the image regression errors - they are based on snapshots that are platform- and backend-dependent).

Creating new bindings

Please provide a test case for every new binding added. See `test/` for examples of unit tests.

Binding to new classes

1. Create your files (e.g. `qclass.h` , `qclass.cc`) from the provided templates `src/template.h` , `src/template.cc`
2. `qclass.*` : search and replace all occurrences of `__Template__` , `__TEMPLATE__` , and `__template__` with the corresponding class name
3. `node-qt.gyp` : Add `qclass.cc` to sources list
4. `qt.cc` : Include `qclass.h`

5. `qt.cc` : Add `QClass::Initialize()` to `Initialize()`

Binding to new methods

1. `qclass.h` : Declare static method as per `Example()` method in `template.h`
2. `qclass.cc` : Implement method as per `Example()` in `template.cc`
3. `qclass.cc` : Expose method to JavaScript via `tpl->PrototypeTemplate()` call in `Initialize()` . Again see `template.cc`.

Common errors

This is a list of common errors when experimenting with Node addons, and their possible solutions:

"Out of memory"

`name` in `NODE_MODULE(name, ...)` does not match target name?

"Unable to load shared library"

`(v8 object)->Set()` called to register a method, but method implementation is missing?

"Segmentation fault"

Tough luck :) Did you forget to `new` a wrapped object?